Instructions for Use of Maple in Mathematics 244

Fall 2005

This document A general introduction to **Maple** can be found in the Math 251 instructions. That document was revised in Fall 2004 to describe features introduced in Maple9, with a slight update in Fall 2005. This document can concentrate on special features that are relevant to Differential Equations. Except for this paragraph, this document for Math 244 is the 2004 version since there has not been sufficient time to study new features of Maple 10. During the Fall 2004 semester, the idea of a **supplementary worksheet** was introduced to encourage experiments with Maple beyond what is needed in the graded report.

Differential Equations Maple has a special set of commands to deal with differential equations in the **DEtools library**. A major reason for including Maple in these courses is to get graphical solutions to problems, and all but the simplest graphs require the **plots library**. Later in the course, the **LinearAlgebra library** is also useful for dealing with **systems** of differential equations. To access these, the **seed files** for this course will begin with `with(plots):with(DEtools):` , and labs 4 and 5 add `with(LinearAlgebra):` to the **setup section** at the start of the worksheet. Note that these commands end with a colon instead of the usual semicolon to suppress output (which would be a list of all objects defined in the library).

The basic command for solving differential equations is `dsolve`. It can find **general solutions involving a parameter** or the **unique solution** of an **initial value problem**. Using **optional arguments**, you can also force Maple to try particular methods. All methods described in the textbook — and many more — are available.

To aid in identifying useful methods, there is a maple function called `odeadvisor`. More information can be found on the (Maple9) help pages **DEtools,odeadvisor** and **dsolve,education**. The tools described there will be an important part of the Maple projects in this course, so you are urged to return to those help pages often.

Despite the wealth of methods for solving differential equations, there are many equations that cannot be solved in closed form. Maple can use **numerical methods** to learn enough about solutions of these equations to graph its solutions. The `DEplot` instruction combines these numerical methods with graphical methods to graph solutions along with a **direction field** that can be determined from the equation.

An example admitting a closed-form solution is the differential equation

$$\frac{d^2y}{dx^2} - y = x^3. \qquad (D)$$

To describe this equation in Maple, we need to communicate that the unknown solution y is a function of x, and to describe the relevant derivatives of y with respect to x. To do this, we write $y(x)$ instead of y when referring to this variable. Since the first derivative is `diff(y(x),x)`, the second derivative is `diff(y(x),x$2)` or `diff(y(x),x,x)`, with similar expressions for higher derivatives, the differential equation $d^2y/dx^2 - y = x^3$ should be expressed to Maple as `diff(y(x),x,x)-y(x)=x^3` . This can be given the name `de0` by using the instruction `de0:=diff(y(x),x,x)-y(x)=x^3` so that the symbol `de0` will refer to this equation. Note that the first $=$ in this line is preceded by a colon, identifying it as an **assignment** symbol, but the second is not, since it represents part of an **equation**. To place other conditions on $y(x)$, we group additional equations together with `de0` inside braces. For example the initial values

$$y(0) = 1, \quad y'(0) = 2 \qquad (I0)$$

can be described to Maple **as an expression sequence** (that can be easily incorporated into different structures) via `ic0:=y(0)=1,D(y)(0)=2;` . Note that a different syntax is used to describe the value of $y'(0)$ in the initial conditions. Initial conditions involving higher derivatives are more complicated, so it can be helpful to set them using the **interactive tools** that are part of Maple9. There is a `dsolve[interactive]` function that is of some help constructing the commands to solve equations and plot solutions. However, it is limited to constructing the input for the `dsolve` command, and uses **the solution** as input to the `plot` command — contrary to the conventions used in our lab projects. It is possible to get the interactive routine to return the Maple command, but this **output** must be **copied** to an **input line** in order to be reproducible. The first time that you paste such output, you are likely to get a warning about an "incompatible format", but you should not be deterred — the output allows itself to be converted to **usable** Maple input. Thus, the tool can help **build** commands, but must be **replaced** by those commands in the final worksheet.

Unfortunately, the various commands that we use to illustrate the solutions of differential equations use different syntax to describe the equations.

The notation for higher derivatives in initial conditions is too awkward to describe here, but it is generated automatically by `dsolve[interactive]`.

For maximum flexibility in using the tools available in Maple, it is a good idea to introduce separate names for the equations and initial conditions, so that the different functions that work with differential equations can be applied using short commands. For example, after defining `de0` and `ic0` as above, a symbolic solution to the equation (D) with the initial conditions $(I0)$ can be found using the command `dsolve({de0,ic0 });` .

The `DEplot` command obtains a graph using a numerical solution. It is written `DEplot(de0, y(x), x = -2 .. 2, [[ic0]], title="First equation");` . The **title** is optional in Maple, but expected in course projects. One curious feature of this command is that initial conditions must be given as a **list of lists**, so our expression sequence needs to be bracketed **twice**. However, a second set of initial conditions

$$y(0) = 1, \quad y'(0) = 0 \tag{I1}$$

could be added to **the same** graph (with a different title) by defining `ic1:=y(0)=1,D(y)(0)=0;` and using `DEplot(de0,y(x),x=-2..2,[[ic0],[ic1]],title="Two solutions");`

Some commands

The simplest command is `dsolve`. As we have seen, it need only be told the equation it is to solve, or given a **set** containing an equation and initial conditions. If a system of equations is given, the functions to be solved for are given as a second argument. Various **options** may also be specified (as described on the help page).

As you learn more about the methods used to solve differential equations, you can use the command `odeadvisor` to see Maple's classification of **single equations** — possibly of high order. For the most part, this classification should agree with the textbook.

The `DEplot` command is useful for showing both the **direction field** that is a **graphical representation** of an equation and solutions satisfying certain initial conditions. The graphs of solutions are obtained using a **numerical solution** of the equations. Examples of its use will appear in the course projects.

There is also an `odeplot` function for showing solution curves that have been previously calculated numerically, and a `dfieldplot` for direction fields, but the `DEplot` command will be used in this course since it is both powerful and easy to use.

Examples

A worksheet containing the instructions with(plots): with(DEtools): de1:= diff(y(x),x) = - y(x) + 1/(1 + exp(x)); ans1:=dsolve(de1,y(x));ans1 denotes an equation s1:=eval(y(x),ans1); the solution can be recovered ans1a:=dsolve(de1,y(0)=-2,y(x)); s1a:=eval(y(x),ans1a); plot(s1a,x=-1..5); dfieldplot(de1,y(x), x=-1..5,y= -6..4); initval:=[y(0)=-2],[y(0)=1]; DEplot(de1, y(x), x=-1..5, initval,y=-6..4);

s1n:=dsolve(de1,y(0)=-2,y(x),numeric); odeplot(s1n,[x,y(x)],-1..5); and other illustrations of topics discussed in these instructions is available as `sample.mw`. You should download the worksheet and step through these examples to see how properties of equations are shown in different commands.

Linear Algebra

This course will use the newer `LinearAlgebra` package instead of the older `linalg` package. Although there are some cases in which it is necessary to refer to functions in this package by long names like `CrossProduct` or `ScalarMultiply`, many shortcuts are available that make Maple look like ordinary mathematical notation. Vectors can be constructed by enclosing a comma-separated sequence of entries in **angle brackets**. These are considered to be **column** vectors. A matrix can be constructed by separating its columns by vertical bars (|) and enclosing the whole expression in angle brackets. A matrix can also be constructed by rows by reversing the way in which the entries are separated.

Multiplication of matrices, or of a matrix times a vector, is denoted by a dot — actually a period.

The `Eigenvectors` function, applied to a matrix give a sequence of two expressions as output: the first is a **vector** of eigenvalues; the second is a **matrix** whose columns are the eigenvectors **in the same order** as the eigenvalues appear in the first part of the result.

A guide to using this package will be part of one of the course projects.

Getting Help From Other Students

The course includes some **laboratory projects** that form a small portion of your grade. Just as with other homework assignments, it is permissible and helpful to discuss these Maple labs with other students. However, the Maple lab reports that you turn in are being graded and will be part of your final course grade. As with all such work, the printed form of the worksheet is expected to be the work of the student who submits it. In particular, the projects include questions that involve your interpretation of results of Maple's computation. The grader will concentrate on the text comments that contain your answers to these questions, that are expected to reflect your **individual understanding** of the topic.

End of Maple instructions document